14｜音效三剑客：变调、均衡器、混响

2021-12-22 冯建元

《搞定音频技术》 课程介绍 >

讲述：冯建元

时长 12:58 大小 11.89M

你好，我是建元。

你是否很好奇《名侦探柯南》里的变声器在现实中能否实现？百万调音师能让本来唱歌跑调的人的歌声变得好听，这到底是用了什么神奇的方法？其实今天我们要讲的音频中的音效，就是为了实现这些变声、修音等特效而设计的一系列的音频处理算法。

为了实现某种特定的效果，音效算法的种类有很多，这节课我主要介绍三种常见的音效算法：变调、均衡器和混响的设计和使用方法。

变调

其实在之前讲弱网对抗部分的时候，我们在做快慢放操作时就使用到了变速不变调算法，这其实是变调算法中的一种用法。在讲算法具体实现之前我们先想一下，变调的物理含义是什么？

之前我们讲语音信号分析的时候说过，不同的人发音的基频是不一样的。而音调和基频是直接相关的，要变调其实就是要改变基频。而基频的本质是一个信号的循环周期的倒数，比如基频是 250Hz，那么当前时间的语音信号就是以 4ms 为周期的信号。**我们要变调，其实就是把这个循环周期进行扩大或者缩小。**

如图 1 所示，如果我们把语音信号的基频提升一倍，或者说提升一个八度或者 12 个半音，那么时域信号（绿线部分，上面为变调前，下面为变调后）语音的波形还是很相似的，只是每一个周期都缩短了一半。

再看看频域信号（红色部分）是不是变得更稀疏了？最下面的那根亮线代表的基频从 250Hz 左右提升到了大约 500Hz 的位置。由于谐波的频率是基频的倍数，所以谐波之间的间隔也变大了。

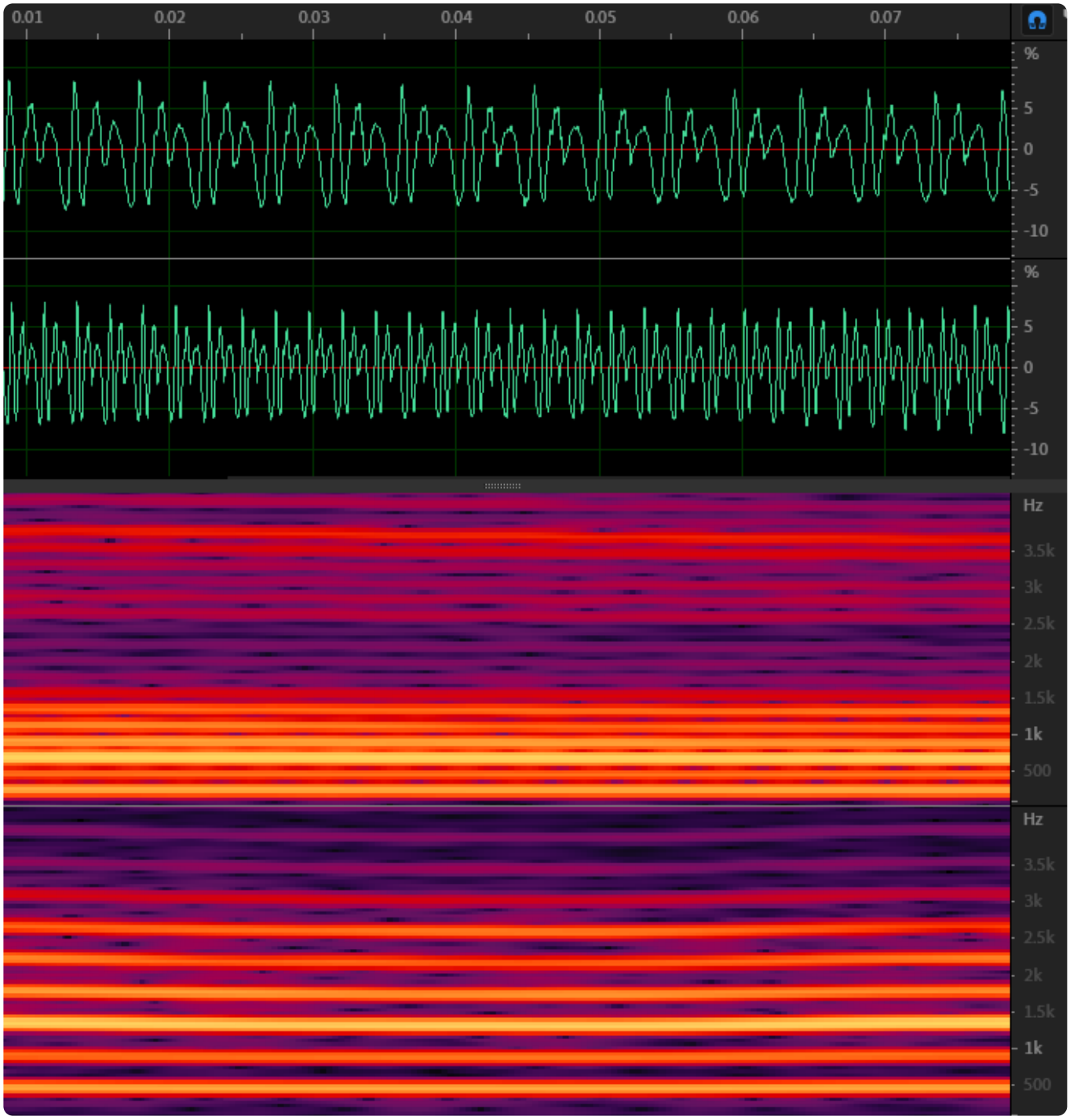


图1 升调后的语音时域和频域图

那么我们是如何实现音调提升的呢？

其实方法很简单，就是**把原来的信号进行重采样，但不改变播放信号的采样率**。简单来说，比如把原来 20ms 的音频每两个点取一个点，然后按照相同的采样率进行播放，这样 10ms 内需要播放原来 20ms 的内容。这样一来，原本的信号循环时间周期就变成了二分之一，从而实现了升调。

但这里有个问题，因为每一段时间内需要播放的音频信号的采样点是固定的。也就是说，通过下采样的方法，音频从原来的 20ms 缩短成了 10ms。直观的感受就是这个人的音调

变高了，但说话的语速也变快了，也就是变速又变调。

但我们想实现的变声只是改变音调，也就是所谓的“**变调不变速**”。其实，实现的方法也很简单，核心思想就是通过把音频中的信号按照一定的规律拼接起来，把音频的长度拉长或者缩短，这就是我们要讲的第一种变调算法 OLA（Overlap-and-Add）。

还记得之前我们讲的 STFT 吗？**其实 OLA 的思想和 STFT 中的 Overlap 的思想很相似。**

如图 2 所示，x 为输入信号、y 为输出信号。OLA 的过程按照图 2 中 b、c、d 的顺序：取一帧 x_m，选择间隔长度 H_a 的下一帧 x_{m+1}，然后把这两帧加窗（汉宁窗）后，以步长 H_s 把两帧重叠、相加在一起。很显然 H_a 和 H_s 的比值就是原始信号和输出信号长度的比值。这样我们就可以把原始音频拼接成不同长度的音频了，然后再经过重采样把音频恢复成和原始音频相同长度的音频再播放，这样就实现了变速不变调。

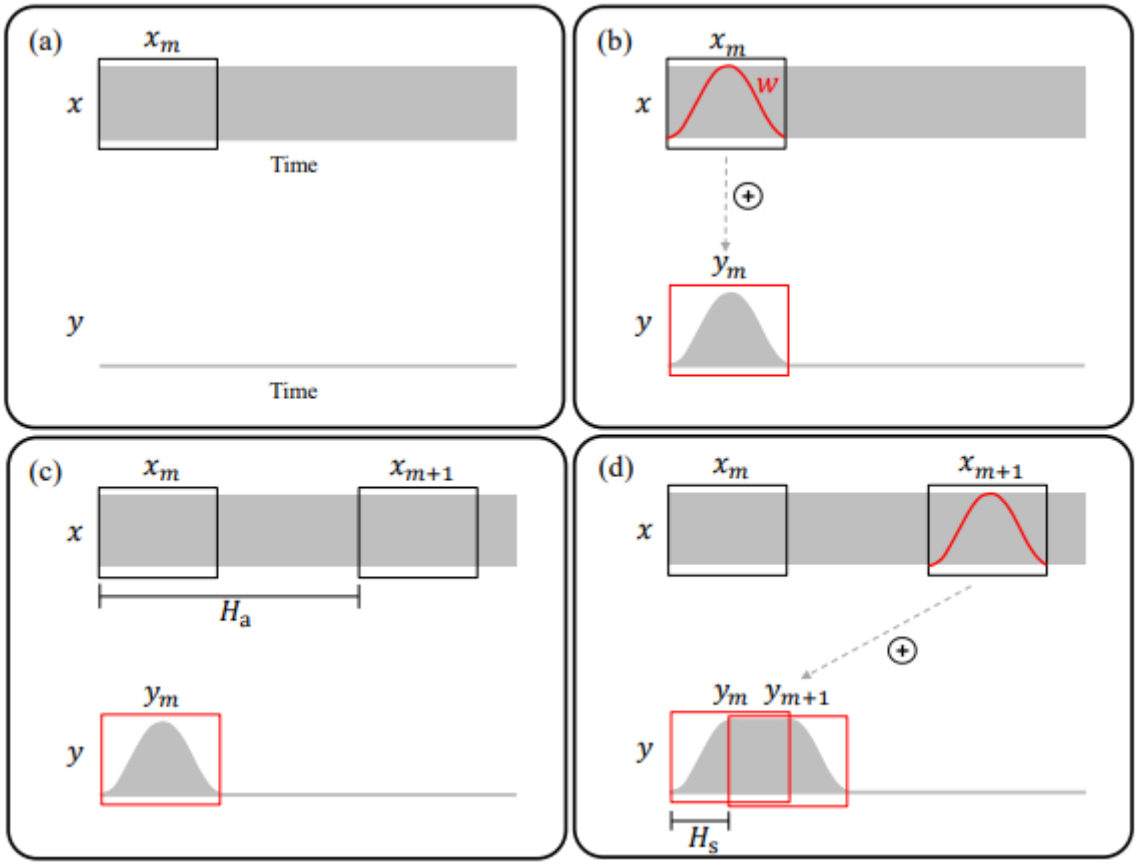


图2 OLA算法的基本原理
图片来源 https://www.mdpi.com/2076-3417/6/2/57

有趣的是，如果不进行重采样直接播放，由于拼接起来的音频没有改变原始语音的基频周期，只是改变了音频的长短，这就实现了我们弱网对抗中用的“变速不变调”的算法。

但是采用这种 Overlap 的方式虽然可以防止连接处的信号产生跳变，但不能保证每一个窗内覆盖的信号都处于周期中的相同相位，或者说**两个窗内信号周期的起始位置不相同**。这就会导致我们常说的“相位失真”。

如图 3 所示，拼接的信号会出现时高时低的现象。所以如果我们能实时的根据信号本身的自相关属性，也就是把信号中相似的两段直接拼接在一起，这样就不会有相位的问题了。基于这样的思想，于是就有了波形相似叠加 WSOLA（Waveform similarity Overlap-Add）算法。

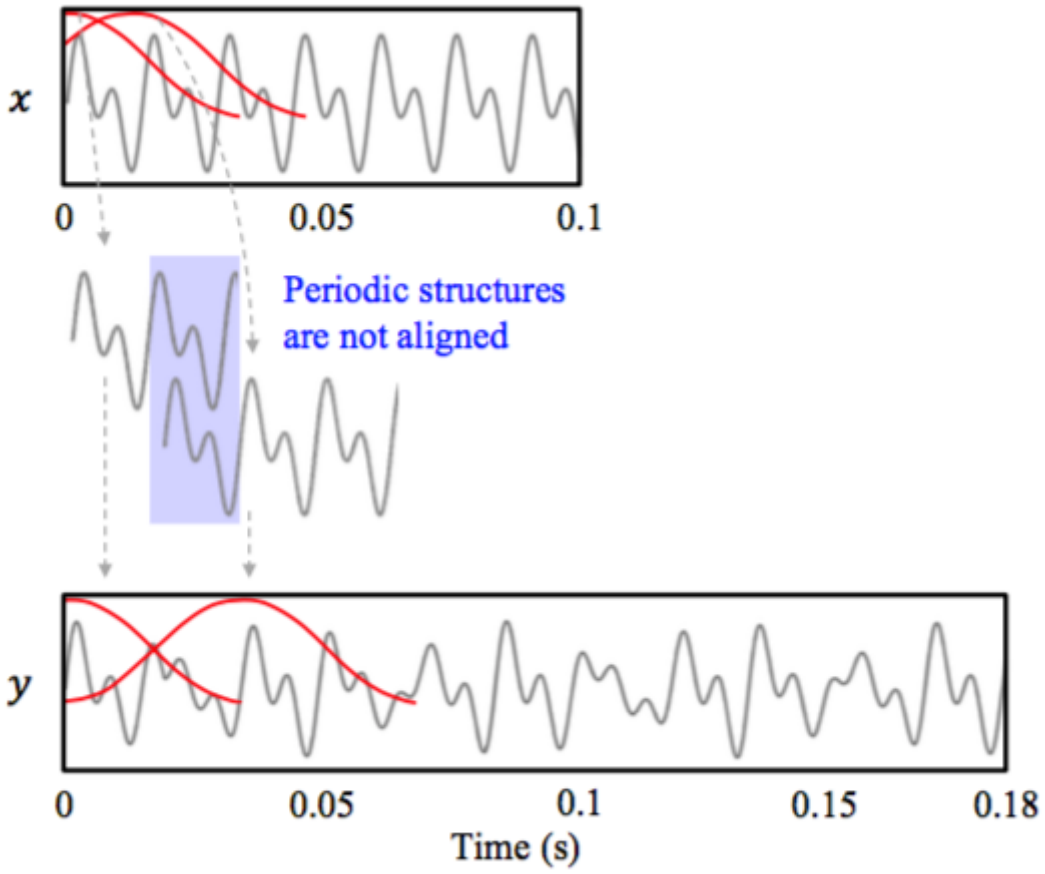


图3 Overlap导致的幅度变化
图片来源 https://www.mdpi.com/2076-3417/6/2/57

WSOLA 算法的计算步骤如图 4 所示，其实相比于 OLA，WSOLA 会在 x_{m+1} 帧的附近寻找和输入信号中如果也移动步长 H_s 的信号 $\tilde{x}_m$ 相似度最高的一段 x'_{m+1} 来做拼接。

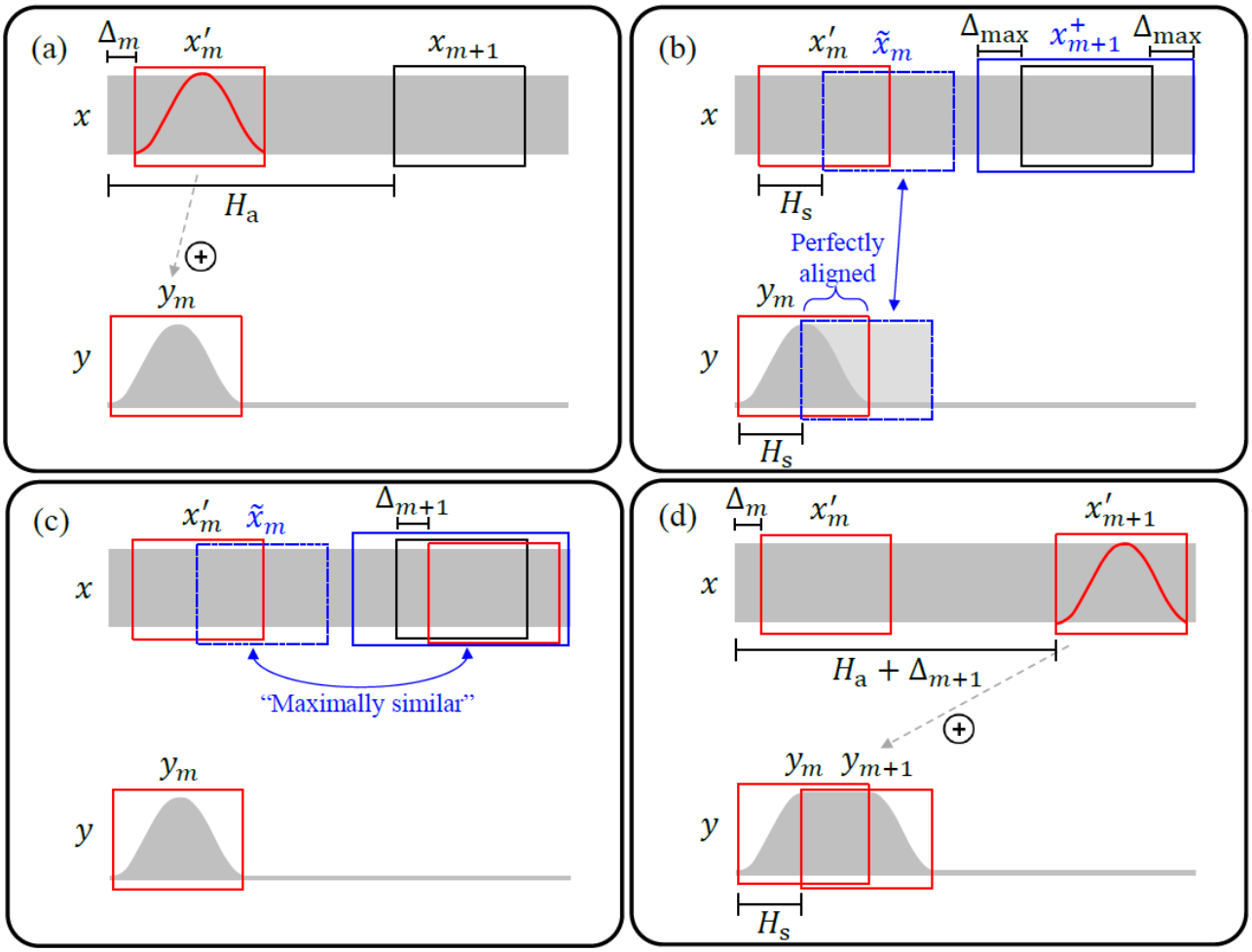


图4 WSOLA的计算步骤
图片来源 https://www.mdpi.com/2076-3417/6/2/57

根据相似性原理，其实 WSOLA 合成出来的变调杂音基本已经没有了，WebRTC 中的快慢放用的就是 WSOLA。但 WSOLA 算法在实时变调中有一个问题，那就是每一帧出现的位置由于需要相似性搜索，所以需要更多的未来帧的信息，也就是说，需要引入更多的延迟。

在时域变调的算法中还有一种 PSOLA（Pitch Synchronous Overlap and Add），顾名思义需要先计算 pitch，然后根据基音周期来改变 Overlap 的大小，这样就直接实现了变调。但是基频检测的鲁棒性没有 WSOLA 的相似性搜索高，所以 PSOLA 的生成可能会出现不稳定的情况，你有兴趣的话可以根据链接自行了解一下。

其实变调算法除了在时域上做拼接，还可以在频域上实现，比如常见的 LSEE-MSTFTM、Phase Vocoder 等算法。

我们看到 WSOLA 等方法通过相似性来寻找拼接对象，但是相似性说到底是通过计算两段时域信号的 MSE（mean squared error）取最小值来得到的，这种方法能尽量保持低频

相位的连续性，但高频信号的相位差异可能不能确保一致。

其中 Phase Vocoder 利用 STFT 中提供的相位信息，在变调扩展的同时会对每个傅里叶频点做相位修正，生成出的音质会比较高，所以在实时变调中常被使用。有兴趣的话，你可以查看文献了解一下。

理解了变调，我们就可以通过算法来改变音频的音调了。比如电影《小黄人大眼萌》中“小黄人”的声音，就是通过变调算法把原本男声的音调提高来实现的。

那么电影《绿巨人》中浩克的那种低沉、怪异的音色又是怎么实现的呢？

均衡器

我们知道每个人都有自己独特的音色，比如有的人声音比较低沉，有的人声音比较清脆。其实对这些音色的感知主要是由于人们在发音时，频谱中不同频段的能量分布不同而导致的。

比如声音低沉的人，可能低频分量比较低，而唱高音、音色饱满的人可能高频的能量也能保持得比较多。而其中最直接的可以改变音色，或者说改变声音在不同频率的能量分布的方法就是 EQ，也就是均衡器（Equalizer）。

那么我们是如何实现一个均衡器来对不同频段的能量进行调整的呢？

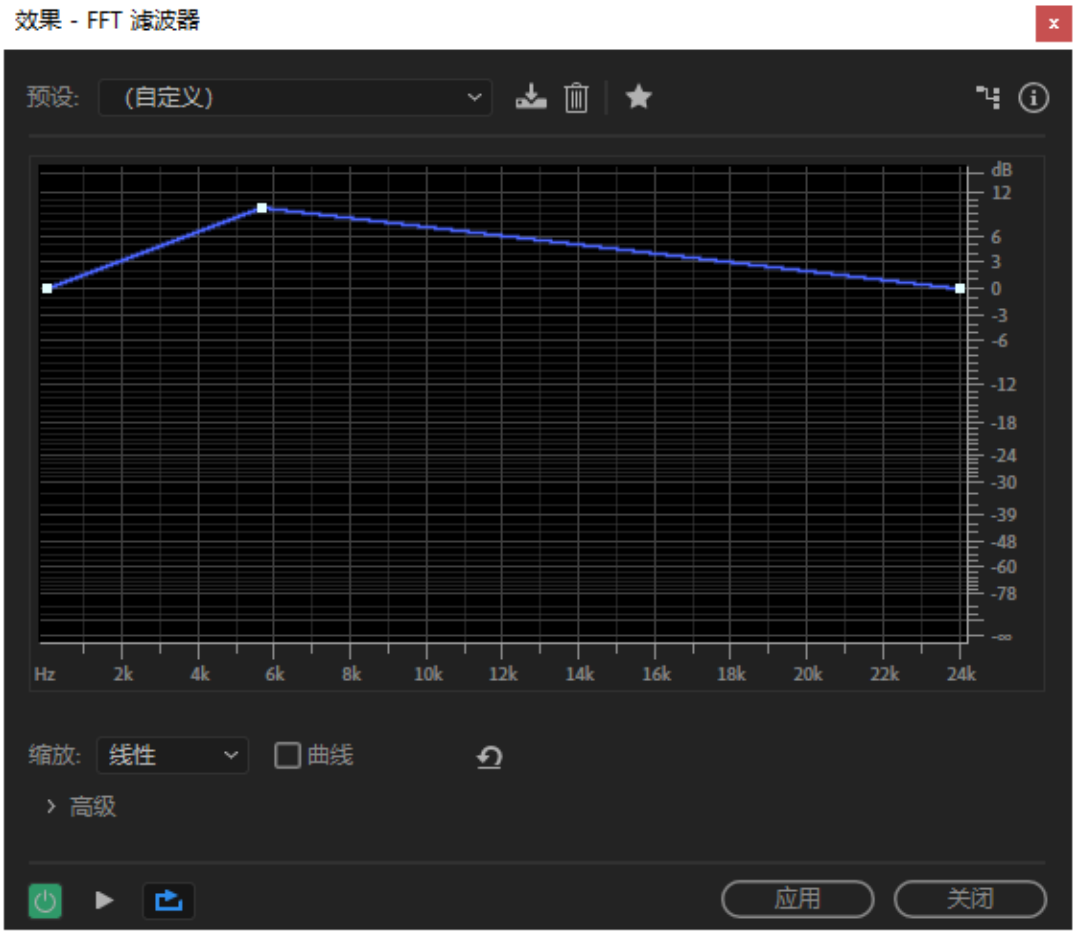
其实均衡器就是一组滤波器，比如常见的高通、低通、带通、带阻等形式。这些可能你之前在大学里的数字信号处理课程里学过。看字面意思应该就可以理解，高通、低通和带通就是让高频、低频或者某个频带的音频保留，而其它的频带都加以削弱，而带阻就是削弱某个频带的音频能量。比如我们觉得人声中齿音太多，想要去齿音，可以在 10kHz ~ 14kHz 左右加一个带通滤波器，削减一下这部分的能量。

音频滤波器有很多种，比如常用的 FIR（Finite Impulse Response）和 IIR（Infinite Impulse Response）Filter。如何根据你想要的频段和增益来设计 FIR 和 IIR 滤波器，其实早已编入了教材，不是信号处理专业的同学可以参考《数字信号处理》这本书。

当然 EQ 的处理经过多年的发展已经有很多通用的滤波器可以选用了，比如椭圆、切比雪夫、巴特沃斯和贝塞尔滤波器等等。如果你想快速实现一个滤波器看看效果也可以直接使

用 Matlab 中的滤波器设计 toolbox 来加速实现进程。

如果不想自己编程实现，也可以利用一些音频处理软件来进行可视化的处理。比如在 Adobe Audition 中，我们可以看到各种常见的 EQ 均衡器。如图 5 所示，由上到下依次为 FFT 滤波器、图形均衡器以及科学滤波器。我们在离线自己做一些音频处理时，可以选择其中的一个或者多个串行使用。



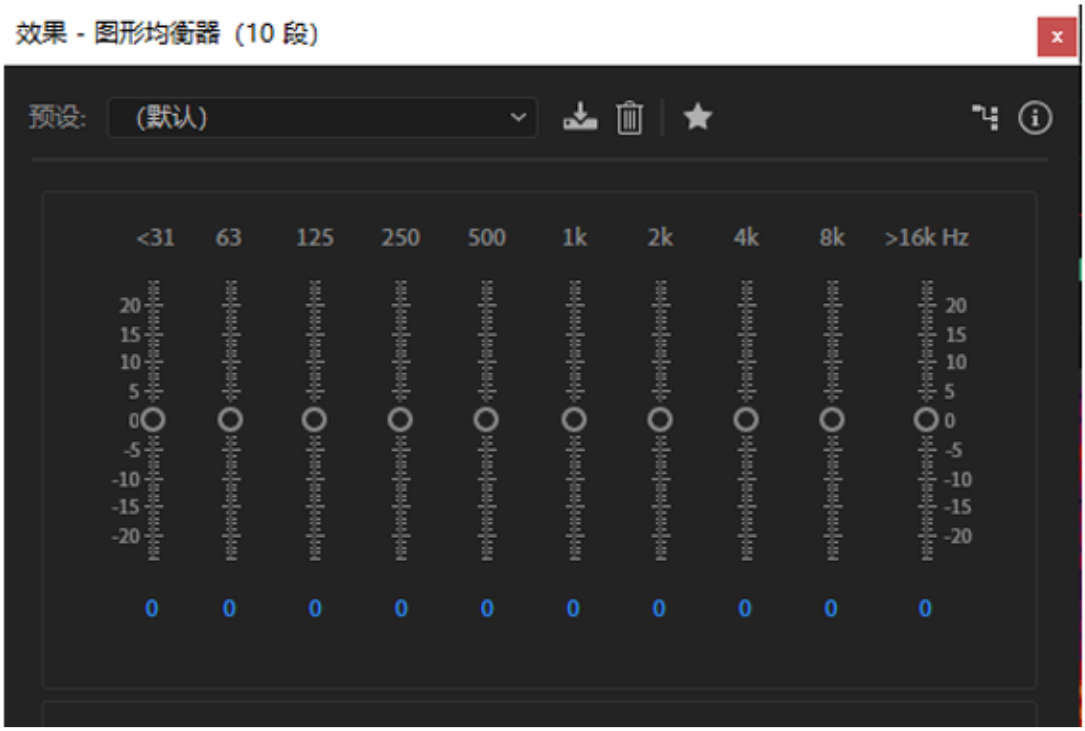


图5 Audition中常见的均衡效果器

混响

好的，知道了如何调节音色，我们再来看看和空间相关的混响。

其实在之前的课程中，我们已经多次提及了混响的概念。在之前讲空间音频时，我们知道可以通过采样的方式或者镜像法得到房间的混响 RIR，这样得到的混响叫做采样混响。采样混响真实，但是不一定好听，并且 RIR 需要和音频信号做卷积才能得到混响信号。当混响时间很长的时候需要的算力也是巨大的。在音乐制作时，为了营造更好的听感经常会使用一些人工混响效果器来产生混响，在实时音效里也可能因为要节省算力而采用人工混响效果器的方式来生成混响。

简单地理解，混响信号可以看作是直达声和许多逐步衰减、不断延迟的回声信号叠加而成的。假设一个衰减系数 a 和延迟 D，那么混响信号 $y(n)$ 可以用下面的等比数列来表示：

$$y(n) = x(n) + ax(n - D) + a^2x(n - 2D) + \dots$$

其中 $x(n)$ 是输入信号，D 为回声的延迟。而这种形式正是梳状滤波器的形式。如图 6 所示，所谓梳状滤波器，其实就是因为它的频率响应呈一个梳子的形状。**梳状滤波常被用来消除某些不需要的谐波，但这里主要是利用了它的拖尾效应。**

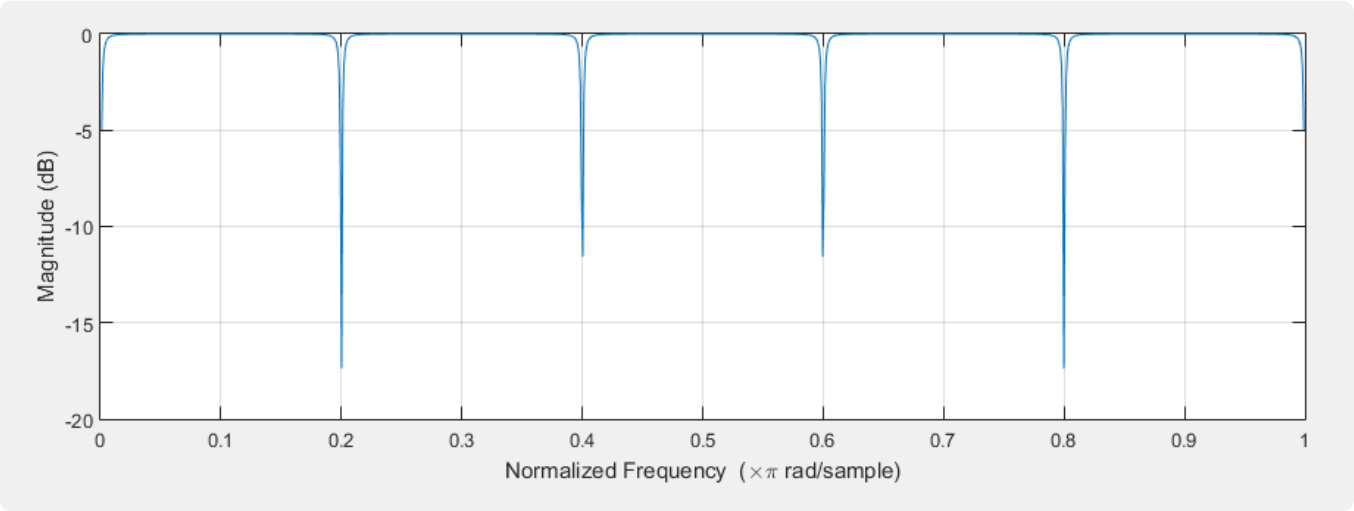


图6 梳状滤波器的频率响应

但也如图 6 所示，梳状滤波器的频谱曲线不平坦，呈现明显的梳状效应。从而对不同的频率成分幅度会产生波动，导致梳状滤波器的拖尾声音带有很强的金属染色效应。而且回声只在延迟为 D 和 D 的倍数的时候出现，这就显得过于稀疏了。所以在梳状滤波器的基础上 Schroeder 使用多个梳状滤波器来解决混响不够密集的问题，然后用全通滤波器（Allpass filter）来消除金属声。

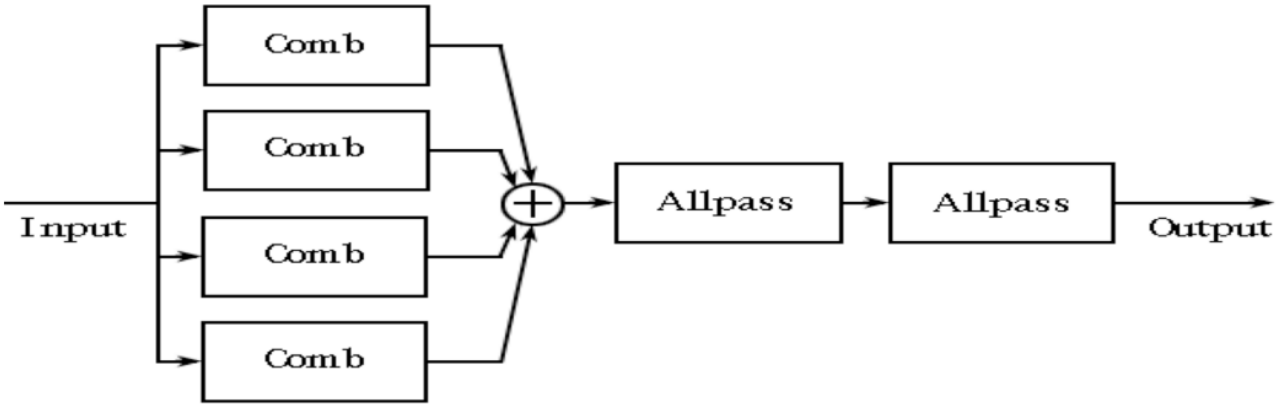


图7 Schroeder混响模型流程
图片来源 http://freeverb3vst.osdn.jp/doc/Sun-Schroeders_Reverberator.pdf

图 7 为 Schroeder 混响模型的结构图，其中每个 Comb 代表一个梳状滤波器。**但 Schroeder 依靠全通滤波器生成的混响依旧不够密集。**

后续 Moorer 又对 Schroeder 模型进行了改良，把混响的生成拆成了直达声、早期混响、晚期混响这三个部分。加入了 FIR 模块来模拟早期混响，用 6 个梳状滤波器和一个全通滤波器来模拟晚期混响，并可以控制各部分的增益。有兴趣的话，你可以通过链接了解一下。

其实 Moorer 模型之后，人们会用各种方式来对混响模型进行改造，现在的混响生成器基本上也都是开放出很多参数，可调节的混响效果器了。比如混响的初始延迟、干湿比、混响 RT60 时间等。这里介绍一个 Freeverb3 开源库，这里有很多基于不同算法的混响效果器实现方法，你可以根据自己的需求了解一下里面的内容。

小结

好的，今天的课程到这里就要结束了，我们来总结一下。

改变声音可以从三个基本的方向来修改，也就是改变音调、音色和混响，分别对应了变调、均衡器和混响这三种算法。

其中变调算法比较常用的有：基于拼接的 OLA、WSOLA 以及频域的 Phase Vocoder 等。而均衡器主要是通过一个滤波器组对不同频段的能量来进行调节，成熟的算法有很多，但大多是 FIR 或者 IIR 滤波器的组合。人工混响则是可以采用多个梳状滤波器的滤波器组，串联全通滤波器来实现，比较常见的有 Scheoeder 和 Moorer 混响模型。

当然音效的种类还远不止这三种，这节课只能算是音效的常见算法的入门。在这里再和你分享一个音效的网页 demo，里面有声网研发的各种声音效果。你可以通过链接感受一下，看看音效还能做出什么不同的效果。

思考题

其实知道了这些音效的基本原理，结合常用的一些开源库或者音频处理软件就可以开始对音频进行音频效果调整了。比如我们之前说的绿巨人的音效，实际上就是通过变调、均衡器和混响来整体调节的。绿巨人需要把原来的声音进行降调，比如降八度，然后可以使用 10 段均衡器，绿巨人的均衡器参数如图 8 所示，然后再加一个小房间的混响就可以实现了。你可以用 Audition 的效果模块自己试着调一调。

图8 绿巨人均衡器参数

期待你的动手实践，过程中遇到任何问题，都可以到留言区中与我交流讨论。也欢迎你把这节课分享给你的朋友，邀请他一同尝试。我们下节课再见！

分享给需要的人，Ta订阅后你可得 20 元现金奖励

生成海报并分享

赞 0 提建议

© 版权归极客邦科技所有，未经许可不得传播售卖。 页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 13 | 如何利用HRTF实现听音辨位？

下一篇 加餐 | 音频技术漫谈之好声音是怎么炼成的？

精选留言 (2)

写留言

大土豆
2021-12-22

这章来的正好，我明天在部门分享音频处理，正好用的上，哈哈。日常的声音处理，各种重采样，男女生变调，音量加减，淡入淡出，变速不变调，变调不变速等，其实ffmpeg+soundtouch都可以完美实现了。我们业务开发基本是游标卡尺式的开发，看字节数组的长度就基本完事了，输入一帧PCM 4096字节，变速不变调2倍的话，输出是2048字节，那就

稳了，如果是变调不变速呢，还是4096字节，如果是48000重采样成8000呢？4096/6的...

展开

2

springXu

2021-12-23

哈，不懂就问。像谍中谍里的，让别人说一段话后，就可以用个麦克风说的话和那位一样了。这个是如何在短时间内做到的？记得网易也有个让使用者读十句句子，就可以用TTS的方式，让这个人帮你读文章了。这又是如何实现的？

共 2 条评论